Advanced Document-Based Application Techniques

Session 118

These are confidential sessions—please refrain from filming or taking pictures.

Application Technologies

Advanced Document-Based Application Techniques

Mark Piccirelli
Software Engineer, Cocoa Frameworks Group

http://developer.apple.com/wwdc2005/

Introduction

- Cocoa has always had a great architecture for document-based applications
- You can do clever things with it if you know how
- Tiger introduced new features you'll want to take advantage of

Prerequisites

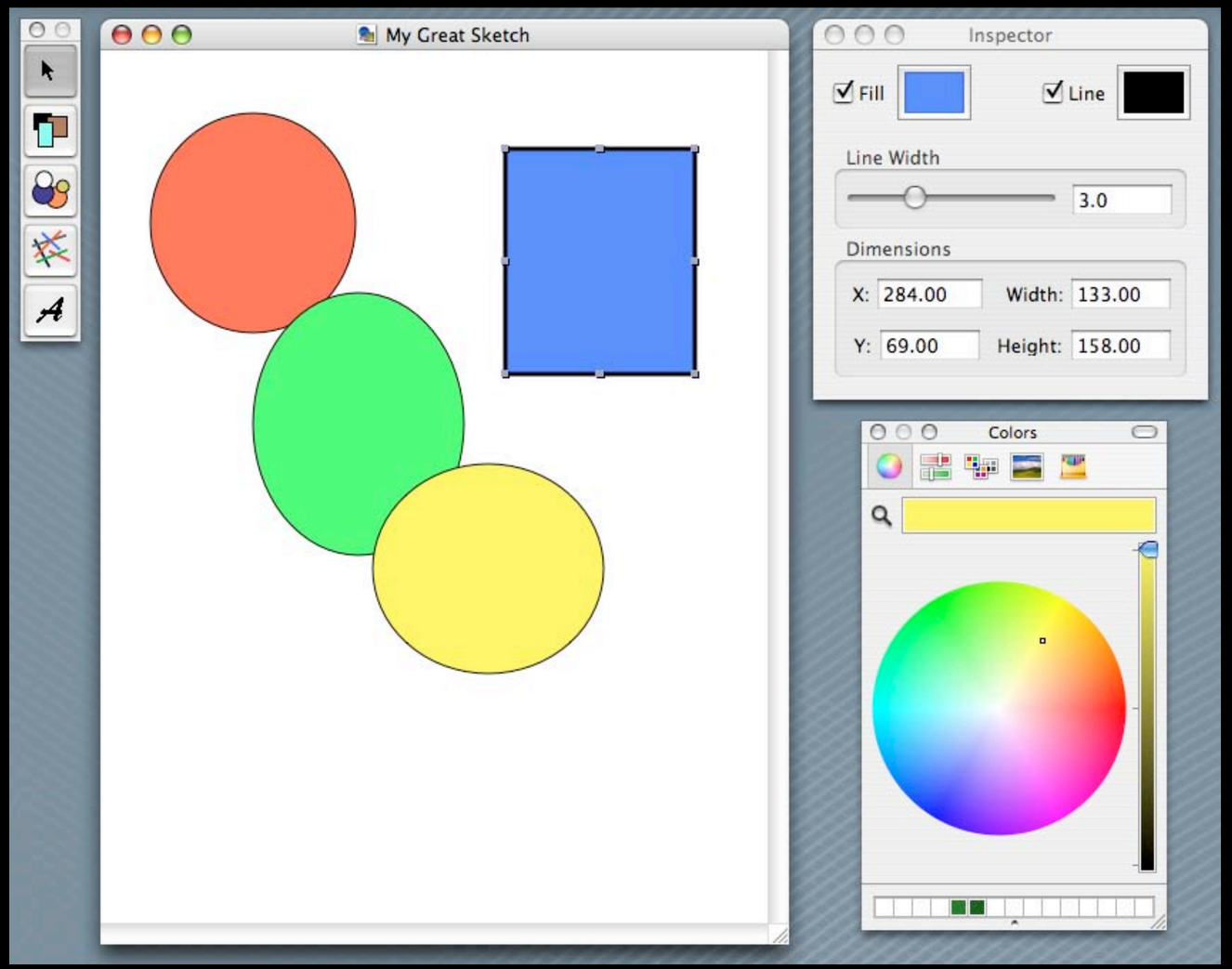
- Experience with Cocoa's document architecture
- Familiarity with Cocoa Bindings is a plus
- We're going to be up to our elbows in NSDocument, NSDocumentController, and NSWindowController

What You'll Learn

- Good error handling
- Supporting new file formats with a plug-in architecture
- Autosaving
- Multiple windows per document
- Multiple documents per window

Today's Sample App

Sketch 2



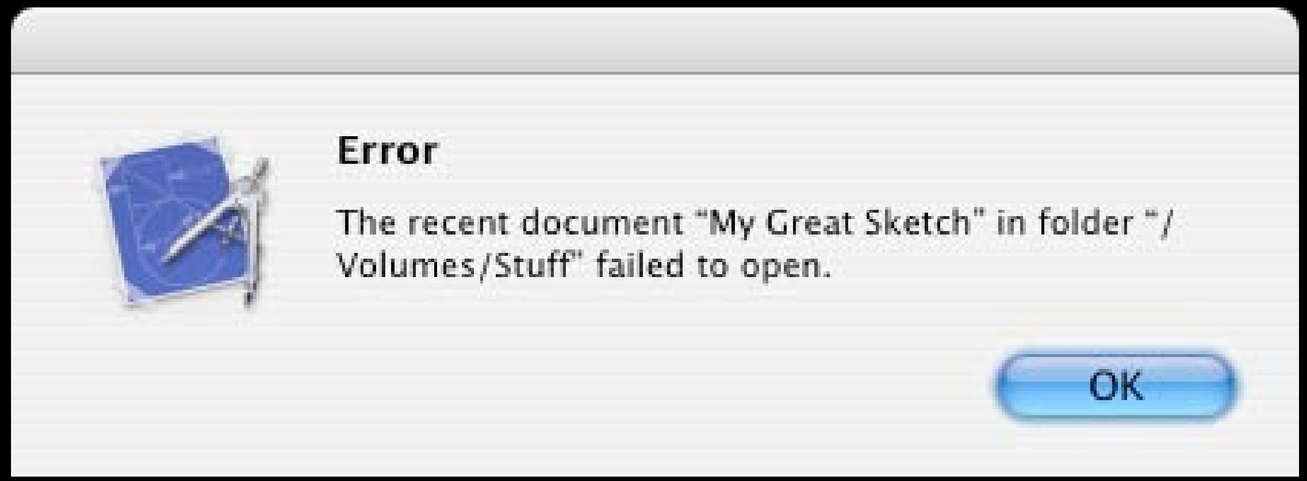
Good Error Handling

New NSError-returning API in Tiger

http://developer.apple.com/wwdc2005/

Good Error Handling

What you got automatically in Panther



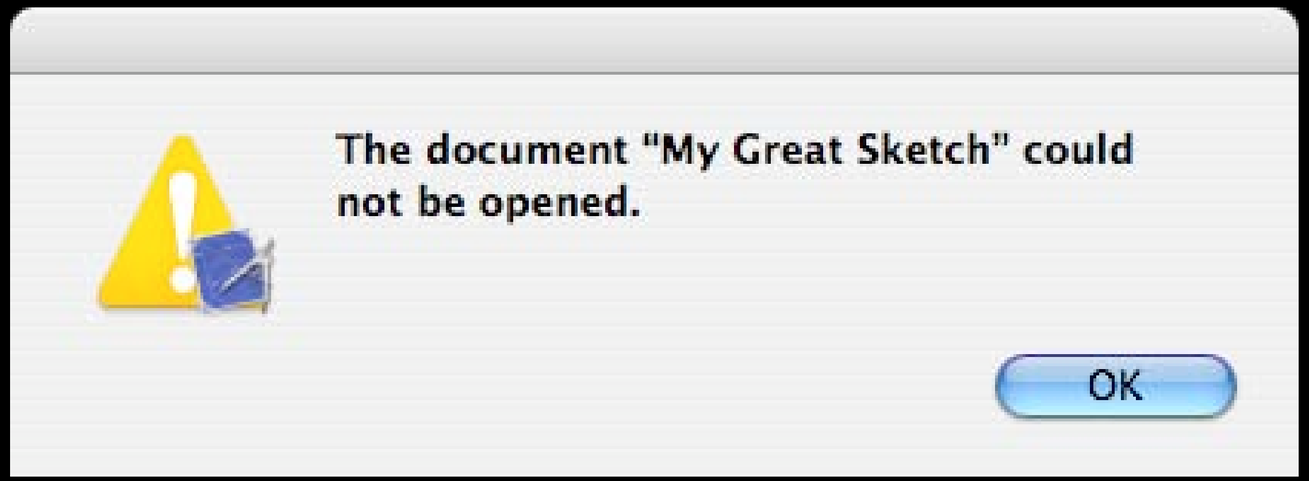
Good Error Handling

What you got automatically in Panther

- There were ways for your app to present its own good error alerts
- It wasn’t easy

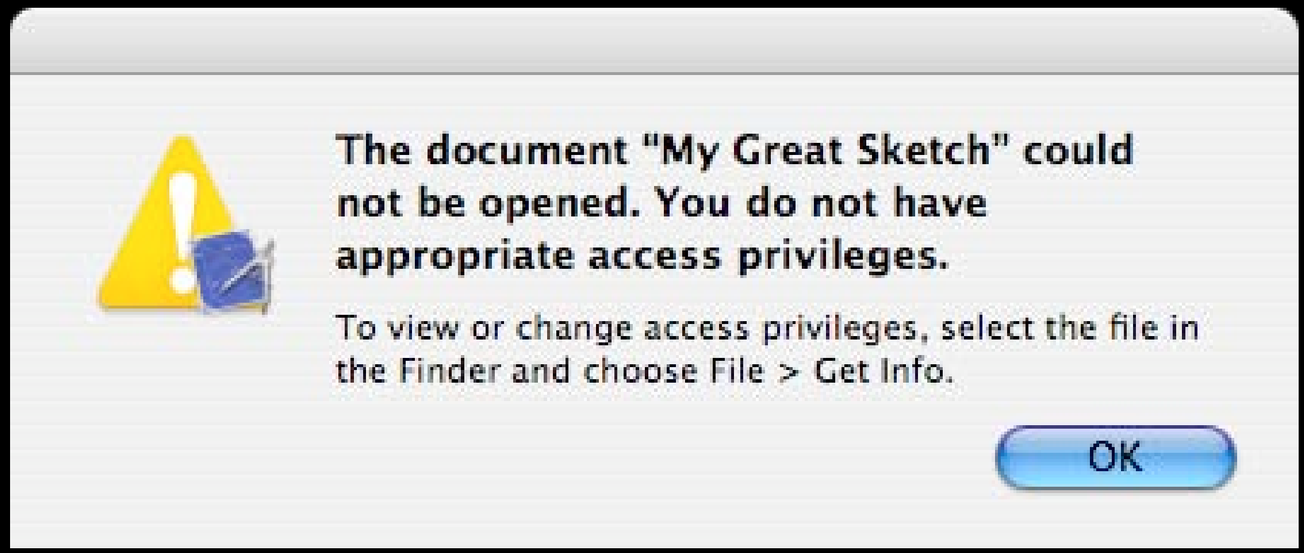
Good Error Handling

What you get in Tiger if you don't update your app



Good Error Handling

What you get in Tiger if you do update your app



Good Error Handling

How do I take advantage?

- Instead of overriding NSDocument methods like these:

  `-readFromFile:ofType:`

  `-writeToFile:ofType:`

- Override methods like these:

  `-readFromURL:ofType:error:`

  `-writeToURL:ofType:error:`

- You have to remove overrides of deprecated NSDocument and NSDocumentController methods

Demo

Let’s update Sketch for better error handling

http://developer.apple.com/wwdc2005/

Good Error Handling

What you just saw

- Override new methods introduced in Tiger
  - Avoid NSDocument's backward binary compatibility code paths
  - Trigger better default error presentation
- Return an NSError when your application code senses the error
  - AppKit presents it to the user for you

Plug-in Architectures

Opened-up document type API in Tiger

http://developer.apple.com/wwdc2005/

Plug-in Architectures

Tiger gives you control over document opening

- If your plug-ins support new file formats you want them to get used
- NSDocumentController used to make that difficult
- Now you can override these NSDocumentController methods:

  `-defaultType` New!

  `-typeForContentsOfURL:error:` New!

  `-documentClassNames` New!

  `-typeFromFileExtension:`

  `-documentClassForType:`

  `-displayNameForType:`

  `-fileExtensionsFromType:`

- You won’t have to override *all* of them

Plug-in Architectures

Tiger gives you control over document saving

- Now you can override these NSDocument methods:
  - `-writableTypesForSaveOperation:` New!
  - `+writableTypes`
  - `+isNativeType`

Plug-in Architectures

Making Sketch 2 extensible with plug-ins

- Sketch 2 uses a new native file format
- We'll support the old format in an importer plug-in
- We'll add an exporter plug-in too

Demo

Let’s add a plug-in architecture to Sketch

http://developer.apple.com/wwdc2005/

Plug-in Architectures

What you just saw

- I’d already added a simple plug-in architecture to Sketch
- We hooked it up to document opening and saving
- AppKit invokes overridable methods when it needs information

Autosaving

Even more new API in Tiger

http://developer.apple.com/wwdc2005/

Autosaving

- Tiger adds support for autosaving to NSDocument
- Helps the user recover from:
  - Application crashes
  - Power failures
  - Things like that
- It’s not on by default
- It’s usually easy to turn on
  - You have to provide a little preferences UI
  - Your document saving code can’t do anything too strange

Demo

Let’s turn on autosaving in Sketch

http://developer.apple.com/wwdc2005/

Autosaving

What you just saw

- We used bindings to easily add UI to support the autosaving preference
- Modified NSDocuments got autosaved automatically
- It didn’t take a lot of code

Multiple Windows per Document

The API has always been there

http://developer.apple.com/wwdc2005/

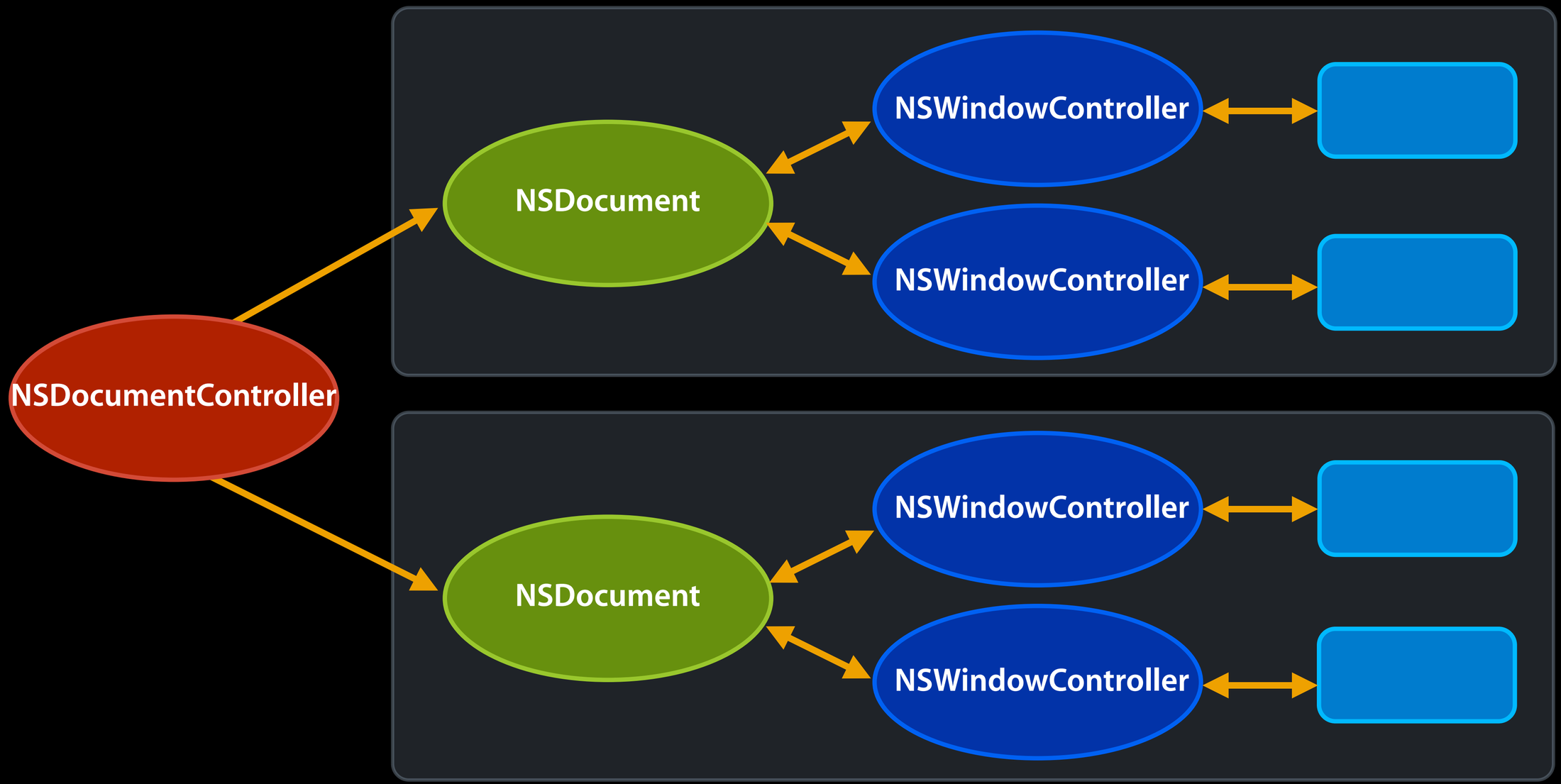
Multiple Windows per Document

NSDocument was designed for this

- Every NSDocument owns one or more NSWindowControllers
- Every NSWindowController has an NSWindow
- Every NSWindowController points back to its NSDocument
- You can add window controllers to already-open documents

Multiple Windows per Document

NSDocument was designed for this



Multiple Windows per Document

What Cocoa does automatically

- Why even tell an NSDocument about NSWindowControllers?
  - A document sends important messages to its window controllers, at the right time

    `-setDocument:`

    `-showWindow`

    `-setDocumentEdited:`

    `-synchronizeWindowTitleWithDocumentName`

    `-close`
  - NSDocument figures out when closing a window means closing the document

Demo

Let’s enable multiple windows per document in Sketch

http://developer.apple.com/wwdc2005/

Multiple Windows per Document

What you just saw

- NSDocument has always supported multiple windows per document out of the box
- We just had to write a little code to open a new window and tell the document about it

Multiple Documents per Window

It’s tricky, but you can do it

http://developer.apple.com/wwdc2005/

Multiple Documents per Window

NSDocument wasn't designed for *this*

- NSDocuments create and own NSWindowControllers
  - Not a good assumption, in this case
- NSDocumentController doesn't know much about NSWindowControllers
  - It just knows to send—makeWindowControllers to each document being opened
  - It doesn't help manage complicated relationships between documents and window controllers

Multiple Documents per Window

NSDocument wasn't designed for this

- That's OK, we just have to:
  - Make up rules for creating new window controllers and associating documents with them
  - Subclass to make a smarter document controller

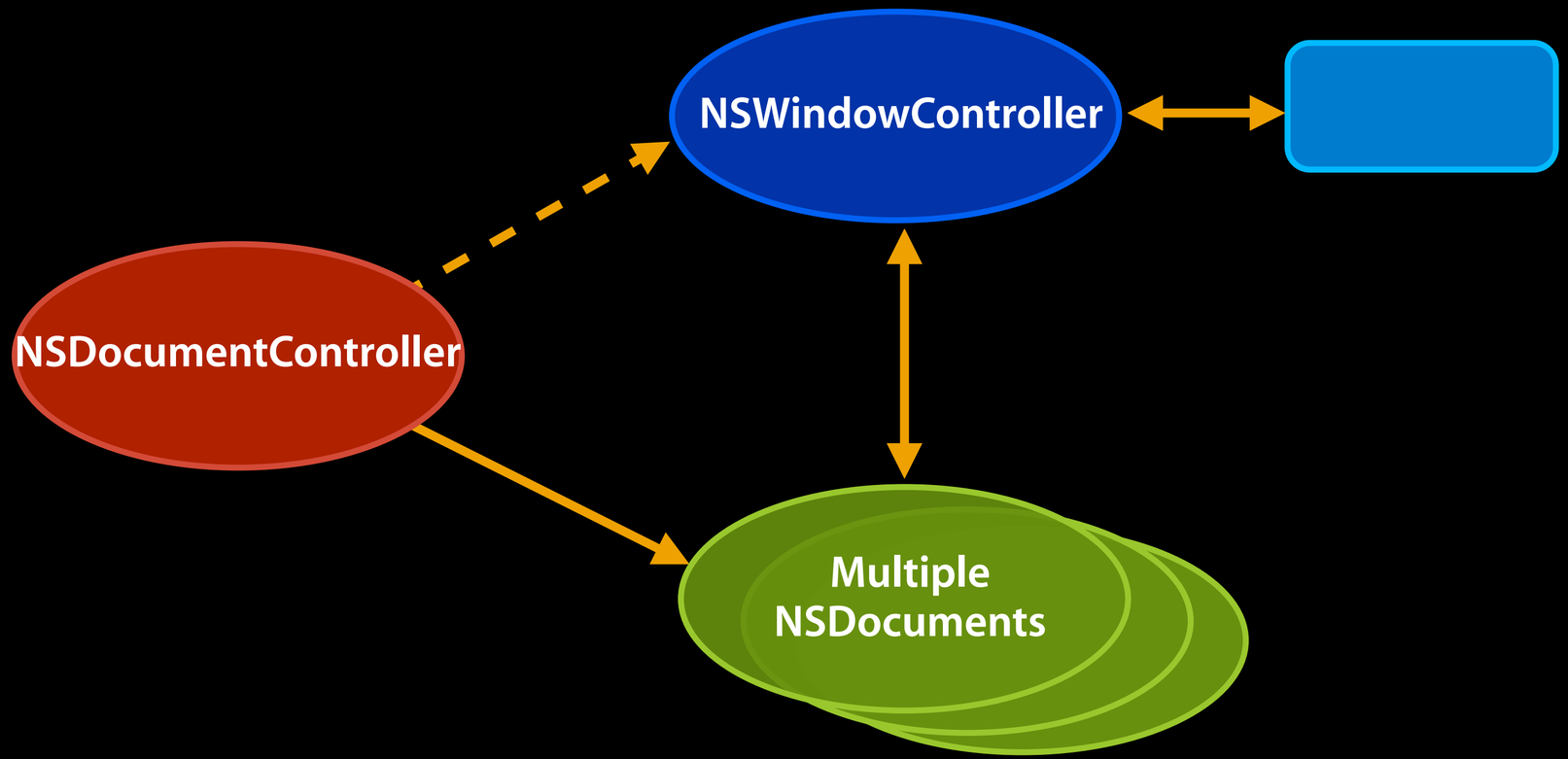
Multiple Documents per Window

NSDocument wasn't designed for this

- Each NSWindowController can point to only one document
- That's OK, we just have to:
  - Know the whats and whens of the messages that are sent back and forth
    - They're all public!
  - Subclass to make a smarter window controller

Multiple Documents per Window

Good object-oriented design makes it possible



Demo

Let's make multiple Sketch documents appear in the same window

http://developer.apple.com/wwdc2005/

Multiple Documents per Window

What you just saw

- “Simple things simple, complex things possible”

More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Who to Contact

Matthew Formica
Cocoa and Developer Tools Evangelist
mformica@apple.com

Related Sessions

Cocoa “Tiger Makeover”	Session 121 Wednesday 2:00PM
Cocoa “Tiger Makeover” Lab	Wednesday 6:30-9PM
Cocoa and Core Data Feedback Forum	Session 139 Friday 9:00AM

Q&A

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference